IN CODE WE TRUST

I-FRAME de VE3PKT

.--. -.- -

Newsletter of the

HAMILTON & AREA PACKET NETWORK

2391 Arnold Crescent,
Burlington, Ontario, Canada, L7P 4J2

Well. Here (finally) is the 4th bulletin of the Hamilton and Area Packet Network (HAPN). The long delay was caused by several things: at the top of the list was procrastination! Next was the implementation and maintainance of the Bulletin Board System. Third was the aquisition of a hard disk for the BBS and the design of a controller for it and last was the discussion about the establishment of a real Station Node in our area. The last 3 items will be dealt with in this newsletter. The first, I leave to your imagination.

Now the table of contents:

1. - a new TIP by Glenn (VE3DSP) for interfacing to a TRS-80

2. - a TIP for the BBS to give you some ideas about computer interfacing (based on a VADCG TIP by J Sprasss)

3. - a description of the Bulletin Board System and its features and the future upgrades to it

4. - the network controller plans

5. -a revised mailing list of members (and hangers-on)

6. - a statement of the current state of communication with other groups

I might add in closing, that the demonstration we put on at the Radio Society of Ontario convention, October 2-4, in Waterloo, Ontario was extremely well received and generated a lot of interest by local amateurs. We have been asked to give further demonstrations to the London Amateur Radio group and a microcomputer club in Toronto.

The Radio Bulletin Board System (RABBS)

The HAPN now has a Bulletin Board System (BBS).

The use we hope will be made of it is for messages relating to communication technology, announcements of new facilities and features (including other group's activities), software distribution with emphasis on communications related routines, maintainance of the mailing list on-line with retrieval by postal code (ZIP), call district, hardware and/or interest keys, and on-line bulletin composition (user submission directly to editor).

The system is based around a home-brew S-100 system. The CPU is a 4 MHz Z-80 on an Ithaca Audio board. Main memory is a 65K S.D. Sales ExpandoRam II, while auxilliary storage is provided by 2 Sugart 801 single sided Double density floppies controlled by a S.D. Sales Versafloppy II board, giving about 800K on-line. A 2K monitor at E000 and a 2K BIOS at F000 are on an Ithaca Prom board. The Radio link is maintained by a VADCG TNC board connected to an RS-232 port on an SMB I/O card with modem control provided by the SMB parrallel port. The system is also accessible over the phone [(416) 335-6620] thru a PMMI MM-103 modem card at 110, 300, 450, 600 and 710 baud. Other cards used locally on the system are a home-brew PROM burner, a S.D. Sales VDB8024 display board and a high resolution graphics board.

The software on the system is CP/M v2.2 while the BBS program is the standard RBBENT27 and RBBMIN27 in Microsoft compiled BASIC. XMODEM is available for file transmission to and from the system as is an extensive HELP system of files.

Current versions of all VADCG TNC software, user's patches, MODEM software, etc. will be maintained on-line. Most files will be in a 'squeezed' form but the 'unsqueezer' will allow users to type out a few lines of files and to expand the files on their own system after transmission in compressed form (lowers communications costs as well).

Now some questions will be answered about the system.

How do I connect to the system?

If you are in the vicinity of Burlington (40 miles) and have a VADCG based system, request a connection to VE3PKT on 146.46 MHz. If the system is available, it will respond to the connection and prompt you for further input (your Call can be aquired automatically by the system).

If you are out of the immediate area, the system can be accessed by phone at area code 416, 335-6620. If the phone is not answered, the system may be in use from the radio link side. Once the system answers, type carriage returns until it responds (usually 2 or 3). This permits the system to determine the baud rate of your transmissions. Use one of the following rates: 110, 300, 450 or 600. The 710 baud rate is only useful if you are within the same exchange because of noise limitations.

At what times is the system available?

We will attempt to maintain the following hours:

-the radio link to the network will be available weekday evenings between 1800 and 2300 Eastern time and 1200 to 2300 on weekends, pending availability of someone to supervise the radio system (with improved software reliability, this may be extended).

-the telephone access will be available from 1800 to 0600 daily and 1200 to 0600 weekends.

What can I do with the system?

Once you have logged on, you are left in control of a CP/M system. You will get a prompt like 'A>' telling you the system is ready. You may type 'HELP' followed by a return to invoke a tree structured help system giving much information about how to run the system, and what is on it. You may type 'MINIRBBS' followed by a return to cause the message system to be loaded. In this routine, you can examine and leave messages for other users or the system operator (SYSOP). You may enter other user areas on the disk (these areas separate sets of files from each other so that all files related to one subject can be stored together) and retrieve or leave files there. These files will contain programs, data and/or text related to communications, applications, or may be submissions to the newsletter so the editor can pull it all together easily. The newsletter itself will be constructed in one user area so you can get parts of it before it is mailed!

You may run the gateway program INTERCOM which will allow access to other links maintained by the system (from phone to radio, radio to phone/DATAPAC, local net to remote net, etc.).

How can I get or leave copies of software on the system?

There are two possibilities here. One is to use the TYPE command and figure out a way for your system to catch the text the BBS sends. This is the simplest but there is no flow control or error correction so it is useful only for small, source files (since binary files cannot be TYPEd out. Another is to use the MODEM software. This permits the user to move single files on or off the system. There are 2 parts to the package:

1. XMODEM- this is a routine resident on the system which can send or receive one file at a time to or from the system. You invoke it by typing (for example) XMODEM S B:FTIP.ASM<return> to send yourself a copy of the file FTIP.ASM from the B: disk, or XMODEM R A:TEST.XXX<return> to leave a file called TEST.XXX on the system's A: drive.

2. MODEM- this routine runs on your system to cooperate with the XMODEM routine on the BBS. It has menu driven input and allows you to specify file names and transfer directions.

You will also require a copy of USQ-15.COM to unsqueeze the larger files once they are on your system. Further information on these routines, and how to get them for yourself can be found in the HELP files on the system.

Network Controller Plans

Discussions with Doug Lockhart are underway to come up with a workable network using the Station Node concept in the Hamilton Burlington Toronto area. Now that Doug has moved to Toronto, we can hope to develop a more ambitious network with his assistance and guidance. One of the projects he and the Vancouver group undertook was the design of a more substantial Station Node Controller card featuring up to 4 simultaneous network operations to be supervised in parallel. His establishment of a Node in Toronto and our plans for a similar one here will allow us the opportunity to test network interconnection as well as routing strategies.

Communications with other groups.

The R.S.O. convention allowed us to make contact with Brett Delmage, VE3XXX, from a group in Ottawa. His network uses the Montreal protocols and is linked to them via packet repeater. He expressed an interest in trying to set up a gateway between their network and ours. Initial thoughts are for a terestrial line of repeaters from here to Ottawa, a plan the Brett and Craig Howie, VE3HWN, are persuing. We hope to have more news about this linking possibility in the near future.

Several other groups have subscribed to the 'I-Frame' recently, among them the Sykes Datatronics Amateur Radio Club in Rochester, New York, who should have no problems contacting us on 2 metres. The PCNet Project also 'logged in' so to speak and we hope to generate a useful exchange there since their interests lie in the area of interconnecting personal microprocessors. There are a lot of non-Amateurs interested in networking, too!

The Latest Mailing List!

The following in a list of people we have on our mailing list. The ones with addresses listed have consented to be identified. We would like you to take the time to fill out the revised 'Brag' sheet and indicate whether you may be identified more fully in future issues. We hope that you might find new aquaintances in the 'Packet Racket' who live close by. We also have plans afoot to further computerize

the list (which also includes details from the Brag sheets) so that it can be searched for entries according to Call district, equipment type, software experience etc.

Robert Sleath,	VE3EFD,
Brian Kennedy,	VS9ASP
AMRAD Corp., 1524 Spring Vale Avenue, McLean, Virginia, 22101.	
John Vander Berg, VE3DVV, RR 2, Group G, Box 14, Mount Hope, Ontario, LOR 1W0.	
Stewart Beal, 2391 Arnold Cres., Burlington, Ontario, L7P 4J2.	VE3MWM
Frank Roberts, 27 Willis Dr., Brampton, Ontario, L6W 1A8.	VE3FAO
Brian Brownlee,	
Max Pizzolato,	VE3DNM
Fulko Hew,	
Glenn Simpson,	VE3DSP
Dan Robertson,	VE3FOV,
W.J.Montgomery,	VE3EC
James Knott,	VE3CVM
Paul Deveraux,	
Michael Connolly, VE3MDC	
Russel S. Milland, VE3FVX	
Dr. George Piasecki,	VE3GUH
Tom Gleason,	VE3MFC (W9ITI)
Keith Witney,	VE6CJM
Clayt Anguish,	VE3LU

Glen Leinweber, VE3DNL

Ray Ewan, VE3HWY

Brian Fox, VE3EBF

Jeff Knight, VE3VA

Al Lightstone, VE3LF

Hugh Pett,

V.A.D.C.G., VE7PKT,

William M. Moran, W4MIB,

Mr. Craig Howie, VE3HWN,

Fred Pluthero, VE3FJW,

Hank S. Magnuski, KA6M,

Douglas Dyment,

B.G. Echols, Jr., WA2NYR/DA2EJ,

BILL GROSKURTH, VE3IRN,

HAL COMMUNICATIONS CORP.,

Jeb Boswell, W1YO,

George Buchanan, VE3IQM,

Dan Howard, VE3BIR,

Dick Tunks, VE3EIE,

Ray Hunter, VE3UR,

Alfred A. Bengel, VE3???,

Jon Duerdoth, VE3BKB,

George Campbell, VE3JSG,

Roger Hay, VE3BTH,

Michael Patrick, VE3FMG,

Ed. Leslie, VE3EHL,

Karl J. Miller, K2MJ,

John T. Varga, WA8ZIA,

Rickey A. Caldwell,

Tom Feeny, W8KOX,

John Gilmore, PCNet Project,

Brett Delmage, VE3JLG,

Sykes Datatronics Amateur Radio Club,

Jack Botner, VE3LNY,

Lloyd R.W. Morgan,

M.G.Reedy, W1BOL,

Charles M Aanonson, Jr.,

Jim Fawns, VE3JAT,

Don Folk, N6AGR,

```
                TITLE   'VADCG TNC - MODULE TIP-TTC (Last Revised 2308 08-Sep-81)'
        ; ***************************************************************************
        ; **   VADCG TERMINAL NODE COMMUNICATIONS PROGRAM  -  MODULE TIP-TTC      **
        ; **          BY DOUG LOCKHART, VE7APU                MAY, 1980            **
        ; ***************************************************************************
        ; LAST CHANGED: JULY 13, 1981

        ; TERMINAL INTERFACE PROGRAM
        ; THIS PROGRAM IS WRITTEN TO RUN IN THE VADCG TERMINAL NODE CONTROLLER.  IT
        ; INTERFACES WITH A NODE COMMUNICATIONS PROGRAM RUNNING AT ADDRESS 0 IN
        ; MEMORY.  THIS VERSION IS WRITTEN TO USE THE 8250 PROGRAMMABLE UART
        ; TO COMMUNICATE WITH A COMPUTER.
        ; THE BASIC FEATURES OF THIS TIP ARE:
        ; NO ECHO OF DATA TO DIGITAL EQUIPMENT
        ; AUTOMATIC TRANSMISSION OF DATA IN BUFFER AFTER DELAY
        ; AUTOMATIC PACKET GENERATION IF DATA IS RECEIVED WITHOUT LF.
        ; CTS FLOW CONTROL FROM DIGITAL EQUIPMENT TO TIP
        ;
        ;       WHEN DSR IS HIGH, CONNECTIONS ARE ALLOWED. WHEN IT FALLS, A
        ;       DISCONNECT IS DONE. DTR IS SET HIGH IF CONNECTED, LOW IF NOT
        ;
        ;       TRANSMIT INTERRUPT CODE HAS BEEN RE-WRITTEN AND INCLUDED
        ;       IN THE DISPATCHER AT MAINLINE LEVEL TO ASSURE CORRECT
        ;       SYNCHRONIZING OF THE HOST. WHEN CONNECTED, EACH TIME THE
        ;       CTS LINE RISES, THE TIP MAY SEND 1 BYTE BUT MUST WAIT
        ;       TILL IT FALLS BEFORE RE-ARMING.
        ; CHANGED 08-SEP-81. MOVE STACK ON LIP SO MUST CHANGE LOCATION OF
        ;       XMITSYNC. ALSO CHANGED DELAY VALUE BEFORE ISSUING A FRAME.
        ;
                MACLIB  LIB85   ; INCLUDE EXTRA 8085 INSTRUCTION SET

        INCTB   MACRO   ?D
                IF      NOT NUL ?D
                MVI     A,?D
                ENDIF
                RST     2
                ENDM

        INCLB   MACRO   ?D
                IF      NOT NUL ?D
                MVI     A,?D
                ENDIF
                RST     3
                ENDM


        ; RAM CONSTANT - CHANGE FOR DIFFERENT RAM LOCATION
1000 =  LORAM   EQU     1000H   ; START OF RAM STORAGE


        ; NON-ZERO STATUS MEANS LINE BUFFER ADDRESS IS IN HL REG.
        ; ZERO STATUS MEANS NO BUFFER IS READY
        NEXTIN  MACRO
                RST     4
                ENDM
```

note: pg 2 and 3 not printed: STd definitions - nothing special SB.

```
                ; CIRCULAR LINE BUFFER VARIABLES

1012 =          LBPE    EQU     CCA+12H ; LINE BUFFER PROCESSING ENTRY
1014 =          CLBE    EQU     CCA+14H ; CURRENT LINE BUFFER ENTRY ADDRESS
1016 =          OLBE    EQU     CCA+16H ; OLDEST LINE BUFFER ENTRY
1018 =          LBIP    EQU     CCA+18H ; LINE BUFFER INPUT POINTER
101A =          LBOP    EQU     CCA+1AH ; LINE BUFFER OUTPUT POINTER

                ; MISCELLANEOUS

1000 =          STAT1   EQU     CCA     ; MAINLINE STATUS BYTE

1003 =          TBOFLO  EQU     CCA+3   ; TERMINAL BUFFER OVERFLOW STATUS
101C =          BUFCOUNT EQU    CCA+1CH ; CURRENT INPUT BUFFER COUNT
101D =          OUTCOUNT EQU    CCA+1DH ; CURRENT BUFFER OUTPUT BYTES REMAINING
1040 =          WAIT    EQU     CCA+40H ; CHARACTER DELAY VALUE
                ;
1100 =          XMITSYNC        EQU     CCA+0100H       ;SYNC AND UNDERWAY FLAG
0001 =          TXUND   EQU     01H     ;TIP TRANSMIT UNDERWAY
0002 =          TXSYNC  EQU     02H     ;TIP MAY TRANSMIT 1 CHARACTER

000D =          CR      EQU     0DH     ; ASCII CARRIAGE RETURN
000A =          LF      EQU     0AH     ; ASCII LINE FEED

1031 =          MODE    EQU     CCA+31H ; MODE OF OPERATION
0080 =          CONNECTED       EQU     80H
0040 =          DISCONNECTING   EQU     40H
0020 =          CONNECTING      EQU     20H
0001 =          ACCEPTCON       EQU     01H     ;ACCEPT CONNECT REQUEST BIT

00FF =          TRUE    EQU     0FFH    ; FOR IF CONDITION TESTS
0000 =          FALSE   EQU     0       ; FOR IF CONDITION TESTS
                ; ***************************************************************
                ; **                    CONFIGURATION EQUATES                  **
                ; **            VALUES CHANGE FOR EVERY CONFIGURATION          **
                ; ***************************************************************

0003 =          FORMAT  EQU     HORST   ; CURRENT CHARACTER FORMAT
0080 =          BAUDRAT EQU     BAUD12  ; CURRENT BAUD RATE

00FF =          CTSFLOW EQU     TRUE    ; IF FLOW CONTROL FROM DIGITAL EQUIPMENT TO TIP
                                ; USING THE CLEAR TO SEND EIA LINE IS IMPLEMENTED.  THE
                                ; DIGITAL EQUIPMENT SHOULD STOP SENDING DATA WHEN CTC DROPS.
                                ; MUTUALLY EXCLUSIVE WITH XONFLOW.  (ALWAYS TRUE AT PRESENT)

0000 =          XONFLOW EQU     FALSE   ; FOR FLOW CONTROL FROM DIGITAL EQUIPMENT TO TIP
                                ; USING XON-XOFF PROTOCOL.  THE DIGITAL EQUIPMENT SHOULD
                                ; STOP SENDING DATA UPON RECEIVING CONTROL-S (DC3) AND
                                ; RESUME SENDING UPON RECEIVING CONTROL-Q (DC1).
                                ; MUTUALLY EXCLUSIVE WITH CTSFLOW. (NOT IMPLEMENTED YET)

000A =          CUSHION EQU     10      ; THE NUMBER OF BYTES THAT MAY STILL BE SENT AFTER
                                ; FLOW CONTROL ACTS TO STOP TRANSFER OF DATA FROM THE
```

```
                                ; DIGITAL EQUIPMENT TO THE TIP.  NOTE THAT SOME EQUIPMENT
                                ; ONLY BREAKS AT END OF LINE.

0C00                    ORG     0C00H   ; WHERE THIS PROGRAM'S EPROM STARTS

                        PAGE
```

```
                ; ENTRY JUMP TABLE

0C00 C3150C             JMP     TIPINIT ; INITIALIZATION ENTRY POINT CALLED BY LIP
0C03 C3360C             JMP     RST55   ; INTERRUPT FROM 8250
0C06 C3060C             JMP     $       ; UNUSED INTERRUPT ENTRY POINT
0C09 C3040D             JMP     DISPATCH        ; TO DISPATCHER ROUTINE
0C0C 0C17564533RIMBUF   DB      12,RIMD,'VE3PKT'        ; CONNECTION BUFFER
0C14 B9         TERMNO  DB      0B9H    ; THIS NODES TERMINAL NUMBER

                TIPINIT:
                        ; SET BAUD RATE IN SERIAL PORT
0C15 3E80               MVI     A,DLAB
0C17 D303               OUT     LCR
0C19 3E80               MVI     A,LOW BAUDRAT
0C1B D300               OUT     DLL             ; BAUD RATE DIVISOR LSB
0C1D 3E00               MVI     A,HIGH BAUDRAT
0C1F D301               OUT     DLM             ; BAUD RATE DIVISOR MSB

                        ; DEFINE CHARACTER FORMAT OF SERIAL DATA
0C21 3E03               MVI     A,FORMAT
0C23 D303               OUT     LCR     ; UPDATE LINE CONTROL REGISTER

                        ; UNMASK INTERRUPTS FROM SERIAL INTERFACE
                        RIM             ; GET CURRENT INTERRUPT MASK IN A
0C25+20                 DB      20H
0C26 E606               ANI     00000110B       ; RESET RST5.5 MASK BIT
0C28 F608               ORI     MSE     ; SET MASK SET ENABLE BIT
                        SIM             ; ENABLE RST5.5 INTERRUPTS
0C2A+30                 DB      30H

                        ; CLEAR OUT RECEIVE BUFFER REGISTER
0C2B DB00               IN      RBR

                        ; ENABLE RECEIVED DATA AVAILABLE AND MODEM INTERRUPTS
0C2D 3E09               MVI     A,ERBFI+EDSSI
0C2F D301               OUT     IER     ; UPDATE INTERRUPT ENABLE REGISTER

                        ; BRING UP RLSD AND CLEAR TO SEND FOR TERMINAL
                        ; RTS = CTS, OUT1 = RLSD
0C31 3E06               MVI     A,OUT1+RTS
0C33 D304               OUT     MCR     ; UPDATE MODEM CONTROL REGISTER

                        ; RETURN TO LIP FOR COMPLETION OF INITIALIZATION
0C35 C9                 RET
                        PAGE
```

```
0C36 F5         RST55:  PUSH    PSW
0C37 E5                 PUSH    H
0C38 D5                 PUSH    D
0C39 C5                 PUSH    B
0C3A DB02               IN      IIR     ; GET INTERRUPT IDENTIFICATION INFOR
0C3C FE04               CPI     IID1    ; IS IT RECEIVED DATA AVAILABLE INTE
0C3E CA7D0C             JZ      RXINT   ; YES, GO TO RECEIVE INTERRUPT ROUTI
0C41 B7                 ORA     A       ; MODEM INTERRUPT?
0C42 C2FE0C             JNZ     EXIT    ; UNKNOWN INTERRUPT, RETURN
                ;
                ; MODEM INTERRUPT. SEE IF CONNECT STATUS CHANGE (VIA DSR)
                ; OR IF DATA SYNCHRONIZATION (VIA CTS)
                ;
0C45 DB06               IN      MSR     ;GET MODEM STATUS
0C47 4F                 MOV     C,A     ;SAVE IT
0C48 E602               ANI     DDSR    ;DELTA DSR?
0C4A CA660C             JZ      MSINT2  ;NO. HOW ABOUT DELTA CTS?
0C4D 79                 MOV     A,C     ;YES. GET THE DSR STATE
0C4E E620               ANI     DSR     ;ISOLATE BIT
0C50 3A3110             LDA     MODE    ;PICK UP LINK STATUS BIT
0C53 CA5E0C             JZ      MSINT1  ;DSR LOW. FORCE A DISCONNECT
0C56 F601               ORI     ACCEPTCON       ;HIGH. ALLOW CONNECTIONS
0C58 323110             STA     MODE    ;TELL THE LIP
0C5B C3FE0C             JMP     EXIT
0C5E E6FE       MSINT1: ANI     0FFH-ACCEPTCON  ;CLEAR THE ACCEPT CONNECTIO
0C60 323110             STA     MODE
0C63 C3970C             JMP     FORCEDISC       ;FORCE A DISCONNECT
0C66 79         MSINT2: MOV     A,C     ;GET STATUS AGAIN
0C67 E601               ANI     DCTS    ;IS IT CTS CHANGE?
0C69 CAFE0C             JZ      EXIT    ;NO. EXIT
0C6C 79                 MOV     A,C     ;GET NEW CTS STATUS
0C6D E610               ANI     CTS
0C6F CAFE0C             JZ      EXIT    ;NOW DOWN. DISPATCH
0C72 3A0011             LDA     XMITSYNC        ;NOW UP. SET THE SYNC BIT
0C75 F602               ORI     TXSYNC  ;TO ENABLE 1 BYTE TO HOST
0C77 320011             STA     XMITSYNC
0C7A C3FE0C             JMP     EXIT    ;AND DISPATCH

0C7D DB00       RXINT:  IN      RBR     ; READ DATA FROM SERIAL PORT
                ;       ANI     7FH     ; TURN OFF HIGH ORDER BIT
0C7F FE18               CPI     18H     ; CONTROL X
0C81 C2920C             JNZ     TESTDIS
0C84 3A3110             LDA     MODE    ;GET LINK STATE
0C87 E6A0               ANI     CONNECTING+CONNECTED    ;ANY ONE CONNECTED
0C89 C2FE0C             JNZ     EXIT    ;YES. IGNORE THE REQUEST
0C8C 3E00               MVI     A,0     ; 0 FOR CONNECT
0C8E F7                 RST     6
0C8F C3FE0C             JMP     EXIT
                TESTDIS:
0C92 FE19               CPI     19H     ; CONTROL Y
0C94 C2A50C             JNZ     OFLOTEST
                FORCEDISC:
0C97 3A3110             LDA     MODE    ;GET LINK STATE
0C9A E6A0               ANI     CONNECTING+CONNECTED    ;ANY ONE CONNECTED
0C9C CAFE0C             JZ      EXIT    ;NO. IGNORE THE REQUEST
```

```
0C9F 3E01             MVI    A,1     ; 1 FOR DISCONNECT
0CA1 F7               RST    6
0CA2 C3FE0C           JMP    EXIT
               OFLOTEST:
0CA5 4F               MOV    C,A     ; SAVE DATA BYTE IN C
0CA6 3A0310           LDA    TBOFLO  ; GET OVERFLOW INDICATOR
0CA9 B7               ORA    A       ; IS THE TERMINAL BUFFER FULL?
0CAA C2FE0C           JNZ    EXIT    ; YES, DON'T DO ANYTHING
0CAD 2A0610           LHLD   OTBE
0CB0 EB               XCHG
0CB1 2A0810           LHLD   TBIP
                      INCTB  1       ; HL = TBIP+1
0CB4+3E01             MVI    A,1
0CB6+D7               RST    2
0CB7 CCF00D           CZ     OVERFLOW        ; INDICATE OVERFLOW IF BUFFER FULL
0CBA CAFE0C           JZ     EXIT    ; EXIT INTERRUPT ROUTINE IF OVERFLOW
0CBD 220810           SHLD   TBIP
0CC0 71               MOV    M,C     ; PUT DATA IN BUFFER
0CC1 210000           LXI    H,0
0CC4 224010           SHLD   WAIT    ; RESET CHARACTER DELAY COUNT
0CC7 211C10           LXI    H,BUFCOUNT
0CCA 34               INR    M       ; INCREMENT BUFFER DATA COUNT
0CCB 79               MOV    A,C     ; GET DATA BYTE IN A
0CCC FE0A             CPI    LF      ; IS IT A LINE FEED?
0CCE CCBA0D           CZ     CLOSE   ; YES, GO TO CLOSE THIS ENTRY
0CD1 3A1C10           LDA    BUFCOUNT        ; GET DATA COUNT
0CD4 FEFA             CPI    250     ; IS IT 250 BYTES OR MORE?
0CD6 D4BA0D           CNC    CLOSE   ; YES, GO TO CLOSE ENTRY
0CD9 3A0310           LDA    TBOFLO
0CDC B7               ORA    A       ; IS TERMINAL BUFFER FULL NOW?
0CDD C2FE0C           JNZ    EXIT    ; YES, NOTHING MORE TO DO
0CE0 2A0610           LHLD   OTBE
0CE3 EB               XCHG
0CE4 2A0810           LHLD   TBIP
                      INCTB  CUSHION ; IS BUFFER CUSHION FREE?
0CE7+3E0A             MVI    A,CUSHION
0CE9+D7               RST    2
0CEA D2FE0C           JNC    EXIT    ; YES, EXIT
               ;
               ;      NO, CHARACTER IN CUSHION, DROP CTS AT HOST TO STOP SENDING
               ;
0CED 3A3110           LDA    MODE    ;GET LINK STATE
0CF0 E6A0             ANI    CONNECTING+CONNECTED    ;ANY ONE CONNECTED TO US
0CF2 3E04             MVI    A,OUT1  ;RLSD LINE FOR HOST BUT NO CTS
0CF4 CAF90C           JZ     FLOW1   ;NO-ONE CONNECTED, LEAVE DTR LINE LOW
0CF7 F601             ORI    DTR     ;SOMEONE THERE, RAISE (OR LEAVE HIGH) DTR LINE
0CF9 D304      FLOW1: OUT    MCR     ; TURN OFF CLEAR TO SEND
0CFB C3FE0C           JMP    EXIT    ; EXIT THE INTERRUPT ROUTINE

0CFE C1        EXIT:  POP    B
0CFF D1               POP    D
0D00 E1               POP    H
0D01 F1               POP    PSW
0D02 FB               EI
0D03 C9               RET
```

```
               DISPATCH:
0D04 010011           LXI    B,XMITSYNC      ;GET SYNC WORD ADDRESS
0D07 F3               DI
0D08 3A3110           LDA    MODE    ;FETCH TNC MODE
0D0B E6A0             ANI    CONNECTING+CONNECTED
0D0D CA220D           JZ     DISPATCHD       ;NO.
0D10 DB04             IN     MCR
0D12 E601             ANI    DTR             ;HAVE WE TOLD HOST ABOUT CONNECTION?
0D14 C22A0D           JNZ    DISPATCHX       ;YES. DONT MESS WITH TXSYNC
0D17 0A               LDAX   B               ;NO, GET SYNC WORD
0D18 E6FD             ANI    OFFH-TXSYNC     ;KILL SYNC BIT TO PRESERVE ALL DATA NOW
0D1A 02               STAX   B
0D1B DB04             IN     MCR             ;FETCH MODEM CONTROL LINE SETTINGS
0D1D F601             ORI    DTR             ;GET BIT TO TELL HOST ABOUT CONNECTION
0D1F C32A0D           JMP    DISPATCHX
               DISPATCHD:
0D22 0A               LDAX   B       ;ALLOW ALL DATA TO GO OUT IF
0D23 F602             ORI    TXSYNC  ;NOT CONNECTED IN ORDER TO
0D25 02               STAX   B       ;KEEP TIP BUFFERS DRAINED
0D26 DB04             IN     MCR
0D28 E6FE             ANI    OFFH-DTR        ;KEEP DTR LOW SO HOST KNOWS NO-ONE CONNECTED
               DISPATCHX:
0D2A D304             OUT    MCR
0D2C FB               EI
0D2D DB05             IN     LSR     ; GET LINK STATUS REGISTER
0D2F E640             ANI    TSRE    ; IS THE TRANSMIT SHIFT REGISTER EMPTY?
0D31 CA740D           JZ     DELAYCHECK      ; NO, THEN GET OUT OF HERE
0D34 0A               LDAX   B       ;GET SYNC WORD
0D35 57               MOV    D,A     ;PRESERVE FOR LATER TEST
0D36 E602             ANI    TXSYNC  ;CAN WE SEND A BYTE?
0D38 CA740D           JZ     DELAYCHECK      ;NO, SKIP
0D3B 7A               MOV    A,D
0D3C E601             ANI    TXUND   ;TRANSMIT ALREADY UNDERWAY?
0D3E C2550D           JNZ    TXCONT  ;YES. CONTINUE WITH TRANSMISSION
               ;
                      NEXTIN         ;NO. TRY TO GET ANOTHER BUFFER
0D41+E7               RST    4
0D42 CA740D           JZ     DELAYCHECK      ;NONE AVAILABLE
0D45 7E               MOV    A,M     ; GET DATA LENGTH IN BUFFER HEADER
0D46 321D10           STA    OUTCOUNT        ; AND SAVE IT FOR INTERRUPT ROUTINE
                      INCLB  3
0D49+3E03             MVI    A,3
0D4B+DF               RST    3
0D4C 221A10           SHLD   LBOP    ; LBOP = OLBE + 3
0D4F F3               DI
0D50 0A               LDAX   B       ;GET SYNC WORD
0D51 F601             ORI    TXUND   ;SET UNDERWAY
0D53 02               STAX   B
0D54 FB               EI
0D55 2A1A10    TXCONT: LHLD  LBOP
                      INCLB  1
0D58+3E01             MVI    A,1
0D5A+DF               RST    3
0D5B 221A10           SHLD   LBOP    ; LBOP = LBOP+1
0D5E 7E               MOV    A,M
```

```
0D5F F3               DI
0D60 D300             OUT    THR     ; OUTPUT DATA AT LBOP
0D62 0A               LDAX   B
0D63 E6FD             ANI    OFFH-TXSYNC     ;KILL THE SYNC BIT
0D65 02               STAX   B
0D66 FB               EI
0D67 211D10           LXI    H,OUTCOUNT
0D6A 35               DCR    M
0D6B C2740D           JNZ    DELAYCHECK
0D6E F3               DI
0D6F 0A               LDAX   B       ;GET SYNC WORD
0D70 E6FE             ANI    OFFH-TXUND      ;KILL TRANSMIT UNDERWAY
0D72 02               STAX   B
0D73 FB               EI
               ;
               ;      CHECK IF IT IS TIME TO FLUSH THE TIP BUFFER TO THE 
               ;      IF NO CHARACTER RECEIVED WITHIN THE LAST 17 MS (BY 
               ;
               DELAYCHECK:
0D74 3A0310           LDA    TBOFLO
0D77 B7               ORA    A
0D78 C0               RNZ
0D79 DB04             IN     MCR     ; GET MODEM CONTROL REGISTER
0D7B E602             ANI    RTS     ; IS CLEAR TO SEND UP?
0D7D C2910D           JNZ    DELAY2  ; YES, SKIP CTS TURN ON TEST
0D80 2A0610           LHLD   OTBE
0D83 EB               XCHG
0D84 2A0810           LHLD   TBIP
                      INCTB  CUSHION ; IS BUFFER CUSHION FREE?
0D87+3E0A             MVI    A,CUSHION
0D89+D7               RST    2
0D8A D8               RC             ; NO, DON'T TURN ON CTS OR DO DELAY
0D8B DB04             IN     MCR
0D8D F602             ORI    RTS
0D8F D304             OUT    MCR     ; TURN ON CLEAR TO SEND AT HOST
               DELAY2:
0D91 3A1C10           LDA    BUFCOUNT        ; GET CURRENT DATA COUNT
0D94 FE00             CPI    0       ; IS COUNT ZERO?
0D96 C8               RZ             ; YES, NOTHING TO CHECK, RETURN
0D97 2A4010           LHLD   WAIT
0D9A 23               INX    H
0D9B 224010           SHLD   WAIT
0D9E 7C               MOV    A,H
0D9F FE10             CPI    10H
0DA1 C0               RNZ
0DA2 CDDE0D           CALL   DISABLERX       ; DISABLE RECEIVER INTERRUPT
0DA5 3A1C10           LDA    BUFCOUNT
0DA8 FE00             CPI    0
0DAA CAE70D           JZ     ENABLERX
0DAD 3A0310           LDA    TBOFLO
0DB0 B7               ORA    A
0DB1 C2E70D           JNZ    ENABLERX
0DB4 CDBA0D           CALL   CLOSE   ; CLOSE OFF THIS ENTRY
0DB7 C3E70D           JMP    ENABLERX        ; ENABLE RECEIVER INTERRUPTS
```

```
                ; SUBROUTINE TO CLOSE OFF TERMINAL BUFFER ENTRY AND PASS
                ; TO CONTROL OF LIP.
                CLOSE:
0DBA 211C10             LXI     H,BUFCOUNT
0DBD 7E                 MOV     A,M
0DBE 3600               MVI     M,0     ; BUFCOUNT = 0
0DC0 2A0410             LHLD    CTBIE
0DC3 77                 MOV     M,A     ; SET UP DATA COUNT IN HEADER
0DC4 2A0610             LHLD    OTBE
0DC7 EB                 XCHG
0DC8 2A0810             LHLD    TBIP
                        INCTB   1
0DCB+3E01               MVI     A,1
0DCD+D7                 RST     2
0DCE 220410             SHLD    CTBIE
0DD1 CCF00D             CZ      OVERFLOW
                        INCTB   1
0DD4+3E01               MVI     A,1
0DD6+D7                 RST     2
0DD7 220810             SHLD    TBIP
0DDA CCF00D             CZ      OVERFLOW
0DDD C9                 RET     ; RETURN TO CALLER

                ; SUBROUTINE TO DISABLE 8250 RECEIVE INTERRUPTS
                DISABLERX:
0DDE F3                 DI
0DDF DB01               IN      IER
0DE1 E6FE               ANI     OFFH-ERBFI
0DE3 D301               OUT     IER
0DE5 FB                 EI
0DE6 C9                 RET

                ; SUBROUTINE TO ENABLE 8250 RECEIVE INTERRUPTS
                ENABLERX:
0DE7 F3                 DI
0DE8 DB01               IN      IER
0DEA F601               ORI     ERBFI
0DEC D301               OUT     IER
0DEE FB                 EI
0DEF C9                 RET

                ; SUBROUTINE TO TURN ON TERMINAL BUFFER OVERFLOW INDICATOR
                OVERFLOW:
0DF0 3EFF               MVI     A,OFFH
0DF2 320310             STA     TBOFLO
0DF5 C9                 RET

0DF6                    END
```

```
             TITLE   'VADCG TERMINAL NODE COMMUNICATIONS PROGRAM - MODULE TIP-TT'
             ; ****************************************************************
             ; **   VADCG TERMINAL NODE COMMUNICATIONS PROGRAM  -  MODULE TIP-TT     **
             ; **         BY DOUG LOCKHART, VE7APU                 MAY, 1980       **
             ; ****************************************************************
             ;       PROGRAM      : TIP6DSP.ASM
             ;       DISK VOLUME : VE3DSP.911
             ;       LAST CHANGED: OCT-18  1981 BY : VE3DVV
             ;
FFFF =       TRUE    EQU     0FFFFH
             ;
FFFF =       PROM    EQU     TRUE
             ;
0000 =       VE3DSP  EQU     NOT TRUE
FFFF =       VE3IAC  EQU     NOT VE3DSP
             ;
             ; ADDED DOWNLINE HEX LOADER      AUGUST 25, 1980
             ;
             ; TERMINAL INTERFACE PROGRAM
             ; THIS PROGRAM IS WRITTEN TO RUN IN THE VADCG TERMINAL NODE CONTROLLER.  IT
             ; INTERFACES WITH A NODE COMMUNICATIONS PROGRAM RUNNING AT ADDRESS 0 IN
             ; MEMORY.  THIS VERSION IS WRITTEN TO USE THE 8250 PROGRAMMABLE UART
             ; TO COMMUNICATE WITH A LOCAL TERMINAL.


                     MACLIB  LIB85

             INCRTB  MACRO   ?D
                     MVI     A,?D
                     RST     2
                     ENDM

             INCRLB  MACRO   ?D
                     MVI     A,?D
                     RST     3
                     ENDM


             ; RAM CONSTANT - CHANGE FOR DIFFERENT RAM LOCATION
1000 =       LORAM   EQU     1000H   ; START OF RAM STORAGE


             ; NON-ZERO STATUS MEANS LINE BUFFER ADDRESS IS IN HL REG.
             ; ZERO STATUS MEANS NO BUFFER IS READY
             NEXTIN  MACRO
                     RST     4
                     ENDM


             ; 8255 PARALLEL I/O EQUATES

0008 =       PORTA   EQU     8       ; PORT A INPUT AND OUTPUT
0009 =       PORTB   EQU     9       ; PORT B INPUT AND OUTPUT
000A =       PORTC   EQU     0AH     ; PORT C INPUT AND OUTPUT
000B =       CONTROL EQU     0BH     ; CONTROL PORT OUTPUT ONLY
```

```
             ; BAUD RATE EQUATES
0004 =       BAUD384 EQU     4       ; DIVISOR FOR 38,400 BAUD
0008 =       BAUD192 EQU     8       ; DIVISOR FOR 19,200 BAUD
0010 =       BAUD96  EQU     16      ; DIVISOR FOR 9600 BAUD
0020 =       BAUD48  EQU     32      ; DIVISOR FOR 4800 BAUD
0040 =       BAUD24  EQU     64      ; DIVISOR FOR 2400 BAUD
0080 =       BAUD12  EQU     128     ; DIVISOR FOR 1200 BAUD
0100 =       BAUD600 EQU     256     ; DIVISOR FOR 600 BAUD
0200 =       BAUD300 EQU     512     ; DIVISOR FOR 300 BAUD
0400 =       BAUD150 EQU     1024    ; DIVISOR FOR 150 BAUD
0476 =       BAUD134 EQU     1142    ; DIVISOR FOR 134.5 BAUD
0573 =       BAUD110 EQU     1395    ; DIVISOR FOR 110 BAUD
0800 =       BAUD75  EQU     2048    ; DIVISOR FOR 75 BAUD
0C00 =       BAUD50  EQU     3072    ; DIVISOR FOR 50 BAUD



             ; 8250 SERIAL I/O EQUATES

             ; REGISTER EQUATES
0000 =       RBR     EQU     0       ; RECEIVE BUFFER REGISTER  (R)
0000 =       THR     EQU     0       ; TRANSMIT HOLDING REGISTER (W)
0001 =       IER     EQU     1       ; INTERRUPT ENABLE REGISTER  (W)
0002 =       IIR     EQU     2       ; INTERRUPT IDENTIFICATION REGISTER (R)
0003 =       LCR     EQU     3       ; LINE CONTROL REGISTER (R/W)
0004 =       MCR     EQU     4       ; MODEM CONTROL REGISTER (R/W)
0005 =       LSR     EQU     5       ; LINE STATUS REGISTER (R/W)
0006 =       MSR     EQU     6       ; MODEM STATUS REGISTER (R/W)
0000 =       DLL     EQU     0       ; DRIVER LATCH (LSB) (W)
0001 =       DLM     EQU     1       ; DRIVER LATCH (MSB) (W)

             ; INTERRUPT ENABLE EQUATES
0001 =       ERBFI   EQU     1       ; ENABLE RECEIVED DATA AVAILABLE INTERRUPT
0002 =       ETBEI   EQU     2       ; ENABLE TRANSMITTER HOLDING REGISTER EMPTY
0004 =       ELSI    EQU     4       ; ENABLE RECEIVER LINE STATUS INTERRUPT
0008 =       EDSSI   EQU     8       ; ENABLE MODEM STATUS INTERRUPT

             ; INTERRUPT IDENTIFICATION EQUATES
0001 =       IPEND   EQU     1       ; '0' IF INTERRUPT PENDING
0002 =       IID0    EQU     2       ; INTERRUPT IDENTIFICATION BIT 0
0004 =       IID1    EQU     4       ; INTERRUPT IDENTIFICATION BIT 1

             ; LINE CONTROL EQUATES
0001 =       WLS0    EQU     1       ; WORD LENGTH SELECT BIT 0
0002 =       WLS1    EQU     2       ; WORD LENGTH SELECT BIT 1
0004 =       STB     EQU     4       ; STOP BIT SELECT
0008 =       PEN     EQU     8       ; PARITY ENABLE
0010 =       EPS     EQU     10H     ; EVEN PARITY SELECT
0020 =       SPTY    EQU     20H     ; STICK PARITY
0040 =       SBRK    EQU     40H     ; SET BREAK
0080 =       DLAB    EQU     80H     ; DRIVER LATCH ACCESS BIT

             ; MODEM CONTROL EQUATES
0001 =       DTR     EQU     1       ; DATA TERMINAL READY
0002 =       RTS     EQU     2       ; REQUEST TO SEND
```

```
0004 =       OUT1    EQU     4       ; OUT1 LINE ON 8250
0008 =       OUT2    EQU     8       ; OUT2 LINE ON 8250
0010 =       LOOP    EQU     10H     ; MODEM LOOP CONTROL BIT

             ; LINE STATUS EQUATES
0001 =       DR      EQU     1       ; DATA READY
0002 =       OE      EQU     2       ; OVERRUN ERROR
0004 =       PE      EQU     4       ; PARITY ERROR
0008 =       FE      EQU     8       ; FRAMING ERROR
0010 =       BI      EQU     10H     ; BREAK INTERRUPT
0020 =       THRE    EQU     20H     ; TRANSMITTER HOLDING REGISTER EMPTY
0040 =       TSRE    EQU     40H     ; TRANSMITTER SHIFT REGISTER EMPTY

             ; MODEM STATUS EQUATES
0001 =       DCTS    EQU     1       ; DATA CLEAR TO SEND
0002 =       DDSR    EQU     2       ; DELTA DATA SET READY
0004 =       TERI    EQU     4       ; TRAILING EDGE RING INDICATOR
0008 =       DRLSD   EQU     8       ; DELTA RECEIVE LINE SIGNAL DETECT
0010 =       CTS     EQU     10H     ; CLEAR TO SEND
0020 =       DSR     EQU     20H     ; DATA SET READY
0040 =       RI      EQU     40H     ; RING INDICATE
0080 =       RLSD    EQU     80H     ; RECEIVE LINE SIGNAL DETECT

0017 =       RIMD    EQU     17H     ; REQUEST INITIALIZATION MODE CONTROL BYTE


             ; CHARACTER FORMAT EQUATES FOR STANDARD EQUIPMENT
0004 =       MOD15   EQU     STB     ; FOR MODEL 15 BAUDOT TTY
000E =       ASR33   EQU     WLS1+PEN+STB    ; FOR MODEL ASR33 TTY
0003 =       PAUL    EQU     WLS1+WLS0       ; FOR VE3IAC SYSTEM
0003 =       GLENN   EQU     WLS1+WLS0       ; FOR VE3DSP SYSTEM
000A =       BOB     EQU     WLS1+PEN        ; FOR BOB'S TERMINAL
001A =       RICHARD EQU     WLS1+PEN+EPS    ; FOR RICHARD'S TERMINAL
0003 =       DBS1    EQU     WLS1+WLS0       ; FOR VE3DVV HOST SYSTEM

0008 =       MSE     EQU     08H     ; MASK SET ENABLE BIT


                     PAGE
```

Packet Radio Network Brag Sheet

REVISED

Name:

Address:

Postal Code:

Call:

Phone: Home:

Office:

Radio: Manufacturer & Model: [ ] Synth [ ] xtal

Frequency range: Power out:

Antenna: Height:

Terminal Node Equipment:

VADCG [ ] Y [ ] N
If no, what?

Modem:

2206/2211 [ ] Y [ ] N
If no, what?

Terminal Equipment:

[ ] Terminal type equipment: [ ] U/L case? [ ] U case only?
[ ] Ctrl chars? [ ] Cursor control?

Type (model) and Features:

[ ] Computer type equipment: Manufacturer & type:

CPU: [ ] Z-80 [ ] 8080 [ ] 8085 [ ] 6800 [ ] 6502 [ ] 1802
[ ] Other (specify):

CPU Clock: ______ MHz

Memory: RAM ______ K Speed _____ nsec

PROM _____ K Speed ______ nsec

ROM ______ K Contents:

Peripherals:

Disks: # drives: Type:

Tapes:

Video Display: Resolution (X,Y): [ ] Color?

Other:

Software:

Operating System: Version:

Experience level: [ ] novice [ ] BASIC
[ ] Assembler [ ] System programmer

Languages commonly used on your system:

over...

Call:

How do you want to use the Network:

[ ] packet video (SSTV)

[ ] games and simulation

[ ] RTTY replacement (gateway to HF)

[ ] CBBS

[ ] on-line Club Bulletin composition

[ ] on-line swap shop classifieds (entry directly to club bulletin)

[ ] message centre

[ ] etc. please specify:

[ ] packet voice

[ ] remote system use

[ ] gateways

[ ] file transfer

[ ] interactive graphics

What features should appear in the newsletters?

[ ] continued bibliography?

[ ] more hardware?

[ ] less hardware?

[ ] philosophical discussions?

[ ] other? (please specify:)

[ ] selected papers?

[ ] more software?

[ ] less software?

Please complete and return to:
Hamilton and Area Packet Network,
2391 Arnold Cres.,
Burlington, Ontario,
L7P 4J2, Canada.

May we release your ADDRESS ( ), HOME PHONE ( ), SYSTEM INFO ( ) [tick if OK]
in our bulletins and Bulletin Board System?
We hope it may put you in contact with others in your neighborhood.

We have decided to run the membership side of the group on a yearly basis to coincide with January 1st. A membership fee of $10.00 is the current rate. For your membership, you will receive all issues of newsletters published in the current year. This will make accounting, etc. easier for us. We hope it won't cause you any problems.
Please update us as to your operation by completing the revised BRAG sheet above. You can send it back to us with your 1982 membership.
Thanks for your cooperation-

73s Stu for the Hamilton and Area Packet Net.

Stu

```
CP/M MACRO ASSEM 2.0    #004    VADCG TERMINAL NODE COMMUNICATIONS PROGRAM - MODULE TIP-TT

                ; COMMON COMMUNICATIONS AREA

                ; CIRCULAR TERMINAL BUFFER VARIABLES

 1000 =         CCA     EQU     LORAM   ; ADDRESS OF BEGINNING OF COMMON COMMUNICATIONS AREA
 1004 =         CTBIE   EQU     CCA+4   ; CURRENT TERMINAL BUFFER INPUT ENTRY
 1006 =         OTBE    EQU     CCA+6   ; OLDEST TERMINAL BUFFER ENTRY
 1008 =         TBIP    EQU     CCA+8   ; TERMINAL BUFFER INPUT POINTER
 100A =         TBOP    EQU     CCA+0AH ; TERMINAL BUFFER OUTPUT POINTER
 100C =         LTBOE   EQU     CCA+0CH ; LAST TERMINAL BUFFER OUTPUT ENTRY
 100E =         CTBOE   EQU     CCA+0EH ; CURRENT TERMINAL BUFFER OUTPUT ENTRY

                ; CIRCULAR LINE BUFFER VARIABLES

 1012 =         LBPE    EQU     CCA+12H ; LINE BUFFER PROCESSING ENTRY
 1014 =         CLBE    EQU     CCA+14H ; CURRENT LINE BUFFER ENTRY ADDRESS
 1016 =         OLBE    EQU     CCA+16H ; OLDEST LINE BUFFER ENTRY
 1018 =         LBIP    EQU     CCA+18H ; LINE BUFFER INPUT POINTER
 101A =         LBOP    EQU     CCA+1AH ; LINE BUFFER OUTPUT POINTER

                ; MISCELLANEOUS

 1000 =         STAT1   EQU     CCA     ; MAINLINE STATUS BYTE

 1003 =         TBOFLO  EQU     CCA+3   ; TERMINAL BUFFER OVERFLOW STATUS
 101C =         BUFCOUNT EQU    CCA+1CH ; CURRENT INPUT BUFFER COUNT
 101D =         OUTCOUNT EQU    CCA+1DH ; CURRENT BUFFER OUTPUT BYTES REMAINING
 1040 =         TSTAT   EQU     CCA+40H ; TIP STATUS WORD

 000D =         CR      EQU     0DH     ; ASCII CARRIAGE RETURN
 000A =         LF      EQU     0AH     ; ASCII LINE FEED
 001B =         ESC     EQU     1BH     ; ASCII ESCAPE

                        PAGE
```

```
CP/M MACRO ASSEM 2.0    #005    VADCG TERMINAL NODE COMMUNICATIONS PROGRAM - MODULE TIP-TT

                ; ************************************************************************
                ; **                   CONFIGURATION EQUATES                            **
                ; **           VALUES CHANGE FOR EVERY CONFIGURATION                    **
                ; ************************************************************************


                ;FORMAT EQU      DBS1    ; CURRENT CHARACTER FORMAT
 0003 =         FORMAT  EQU      GLENN   ; CURRENT CHARACTER FORMAT
                ;BAUDRAT         EQU     BAUD192 ; 19,200 BAUD
 0010 =         BAUDRAT EQU      BAUD96  ; 9600 BAUD
 0061 =         ADDRES1 EQU      61H     ;TERMINAL NODE ADDRESS (VE3DSP)
 00BD =         ADDRES2 EQU      0BDH    ;TERMINAL NODE ADDRESS (VE3IAC)
 0073 =         ADDRES3 EQU      73H     ;TERMINAL NODE ADDRESS (VE3DVV)

                PAGE
```

```
CP/M MACRO ASSEM 2.0    #006    VADCG TERMINAL NODE COMMUNICATIONS PROGRAM - MODULE

               ;===================== IF  PROM   ========================
               IF PROM
 0800                  ORG     800H    ; WHERE THIS PROGRAM'S EPROM STARTS

 0800 =        EPROM   EQU     $

               ; ENTRY JUMP TABLE

 0800 C33D09           JMP     OPTION  ;INITIAL OPTION
 0803 C30318           JMP     RAM+3
 0806 C30618           JMP     RAM+6
 0809 C30918           JMP     RAM+9
               ENDIF


               ;===================   RUN IN RAM  ======================
               IF NOT PROM
                       ORG     1800H   ; WHERE THIS PROGRAM'S EPROM STARTS

               ; ENTRY JUMP TABLE

                       JMP     TIPINIT ; INITIALIZATION ENTRY POINT CALLED BY NCP
                       JMP     RST55   ; INTERRUPT FROM 8250
                       JMP     $       ; UNUSED INTERRUPT ENTRY POINT
                       JMP     DISPATCH        ; TO DISPATCHER ROUTINE
               ENDIF


               ;       USE PROPER CALL SIGN HERE (6 CHARACTERS)

               IF VE3DSP
               RIMBUF  DB      12,RIMD,'VE3DSP'        ; CONNECTION BUFFER
               TERMNO  DB      ADDRES1 ; THIS NODES TERMINAL NUMBER
               ENDIF

               IF VE3IAC
 080C 0C17564533RIMBUF DB      12,RIMD,'VE3IAC'        ; CONNECTION BUFFER
 0814 BD       TERMNO  DB      ADDRES2 ; THIS NODES TERMINAL NUMBER
               ENDIF



               TIPINIT:

               IF NOT PROM
                       CALL    FLASH   ;FLASH 6 TIMES IF RUNNING IN RAM
               ENDIF
                       ; SET BAUD RATE IN SERIAL PORT
 0815 3E80             MVI     A,DLAB
 0817 D303             OUT     LCR
 0819 3E10             MVI     A,LOW BAUDRAT
 081B D300             OUT     DLL             ; BAUD RATE DIVISOR LSB
 081D 3E00             MVI     A,HIGH BAUDRAT
 081F D301             OUT     DLM             ; BAUD RATE DIVISOR MSB
```

```
CP/M MACRO ASSEM 2.0    #007    VADCG TERMINAL NODE COMMUNICATIONS PROGRAM - MODULE

                ; DEFINE CHARACTER FORMAT OF SERIAL DATA
 0821 3E03      MVI     A,FORMAT
 0823 D303      OUT     LCR     ; UPDATE LINE CONTROL REGISTER

                ; UNMASK INTERRUPTS FROM SERIAL INTERFACE
                RIM             ; GET CURRENT INTERRUPT MASK IN A
 0825+20        DB      20H
 0826 E606      ANI     00000110B       ; RESET RST5.5 MASK BIT
 0828 F608      ORI     MSE     ; SET MASK SET ENABLE BIT
                SIM             ; ENABLE RST5.5 INTERRUPTS
 082A+30        DB      30H

                ; CLEAR OUT RECEIVE BUFFER REGISTER
 082B DB00      IN      RBR

                ; ENABLE RECEIVED DATA AVAILABLE INTERRUPTS
 082D 3E01      MVI     A,ERBFI
 082F D301      OUT     IER     ; UPDATE INTERRUPT ENABLE REGISTER

                ; BRING UP DSR, RLSD AND CLEAR TO SEND FOR TERMINAL
 0831 3E07      MVI     A,DTR+OUT1+RTS
 0833 D304      OUT     MCR     ; UPDATE MODEM CONTROL REGISTER

                ; RETURN TO LIP FOR COMPLETION OF INITIALIZATION
 0835 C9        RET
                PAGE
```

```
0836 F5         RST55: PUSH   PSW
0837 E5                PUSH   H
0838 D5                PUSH   D
0839 C5                PUSH   B
083A DB02              IN     IIR     ; GET INTERRUPT IDENTIFICATION INFORMATION
083C FE04              CPI    IID1    ; IS IT RECEIVED DATA AVAILABLE INTERRUPT?
083E CA4908            JZ     RXINT   ; YES, GO TO RECEIVE INTERRUPT ROUTINE
0841 FE02              CPI    IID0    ; IS IT TRANSMIT BUFFER EMPTY
0843 CA0309            JZ     TXINT   ; YES, GO TO TRANSMIT INTERRUPT ROUTINE
0846 C3FD08            JMP    EXIT    ; UNKNOWN INTERRUPT, RETURN

               RXINT:
               ;  TEST FOR ESCAPE MODE
0849 214010            LXI    H,TSTAT ; GET TIP STATUS
084C 3E80              MVI    A,80H
084E AE                XRA    M
084F FA8908            JM     TESC    ; NOT ESCAPE MODE
0852 77                MOV    M,A     ; UPDATE STATUS
0853 DB00              IN     RBR     ; READ IN BYTE
0855 4F                MOV    C,A
0856 FE30              CPI    '0'     ; SYSTEM RESET ?
0858 CA0000            JZ     0
085B FE31              CPI    '1'     ; TEST FOR SEND FRAME
085D C26B08            JNZ    CTEST
0860 211C10    SENDFR: LXI    H,BUFCOUNT      ;CHECK FOR ZERO LENGTH
0863 7E                MOV    A,M
0864 B7                ORA    A       ; SET FLAGS
0865 CAFD08            JZ     EXIT
0868 C3D808            JMP    SEND
086B FE32      CTEST:  CPI    '2'     ; TEST FOR CONNECT
086D C27608            JNZ    DCON
0870 3E00              MVI    A,0     ; 0 FOR CONNECT
0872 F7                RST    6
0873 C3FD08            JMP    EXIT
0876 FE33      DCON:   CPI    '3'     ; TEST FOR DISCONNECT
0878 C28108            JNZ    TRAPT
087B 3E01              MVI    A,1     ; 1 FOR DISCONNECT
087D F7                RST    6
087E C3FD08            JMP    EXIT
0881 FE34      TRAPT:  CPI    '4'     ; DO TRAP ?
0883 CA2400            JZ     24H
0886 C39808            JMP    STATUS
0889 DB00      TESC:   IN     RBR     ; INPUT BYTE
088B 4F                MOV    C,A
088C FE1B              CPI    ESC
088E C29808            JNZ    STATUS
0891 3E80              MVI    A,80H   ; SET ESCAPE SEQUENCE
0893 B6                ORA    M
0894 77                MOV    M,A
0895 C3FD08            JMP    EXIT
0898 3A0310    STATUS: LDA    TBOFLO  ;GET OVERFLOW STATUS
089B B7                ORA    A
089C C26008            JNZ    SENDFR  ;IF BUFFER FULL, TRY TO SEND IT
089F 2A0610            LHLD   OTBE
08A2 EB                XCHG
```

```
08A3 2A0810            LHLD   TBIP
                       INCRTB 1       ; ADVANCE INPUT POINTER
08A6+3E01              MVI    A,1
08A8+D7                RST    2
08A9 CA6008            JZ     SENDFR
08AC 220810            SHLD   TBIP    ; POINTER UPDATED
08AF 79                MOV    A,C
08B0 77                MOV    M,A
08B1 211C10            LXI    H,BUFCOUNT
08B4 34                INR    M
08B5 DB05      TXWAIT: IN     LSR     ; ECHO CHAR BACK
08B7 E620              ANI    THRE
08B9 CAB508            JZ     TXWAIT
08BC 79                MOV    A,C
08BD D300              OUT    THR
08BF 2A0810            LHLD   TBIP    ;ONLY ONE BYTE LEFT?
                       INCRTB 2
08C2+3E02              MVI    A,2
08C4+D7                RST    2
08C5 DACB08            JC     OFLO
08C8 C2D008            JNZ    OK
08CB 3EFF      OFLO:   MVI    A,OFFH  ;SET OVERFLOW
08CD 320310            STA    TBOFLO
08D0 211C10    OK:     LXI    H,BUFCOUNT      ;CHECK FRAME LENGTH
08D3 7E                MOV    A,M     ; CHECK FRAME LENGTH
08D4 B7                ORA    A
08D5 F2FD08            JP     EXIT
08D8 3600      SEND:   MVI    M,0     ; BUFCOUNT = 0
08DA 2A0410            LHLD   CTBIE
08DD 77                MOV    M,A
08DE 2A0810            LHLD   TBIP
                       INCRTB 1
08E1+3E01              MVI    A,1
08E3+D7                RST    2
08E4 220410            SHLD   CTBIE
08E7 C2EF08            JNZ    SKIP
08EA 3EFF              MVI    A,OFFH
08EC 320310            STA    TBOFLO
               SKIP:   INCRTB 1
08EF+3E01              MVI    A,1
08F1+D7                RST    2
08F2 220810            SHLD   TBIP
08F5 C2FD08            JNZ    EXIT
08F8 3EFF              MVI    A,OFFH
08FA 320310            STA    TBOFLO
08FD C1        EXIT:   POP    B
08FE D1                POP    D
08FF E1                POP    H
0900 F1                POP    PSW
0901 FB                EI
0902 C9                RET


0903 2A1A10    TXINT:  LHLD   LBOP
                       INCRLB 1
0906+3E01              MVI    A,1
```

```
0908+DF                RST    3
0909 221A10            SHLD   LBOP    ; LBOP = LBOP+1
090C 7E                MOV    A,M
090D D300              OUT    THR     ; OUTPUT DATA AT LBOP
090F 211D10            LXI    H,OUTCOUNT
0912 35                DCR    M
0913 C2FD08            JNZ    EXIT
0916 3E01              MVI    A,ERBFI
0918 D301              OUT    IER     ; DISABLE TRANSMITTER INTERRUPTS
091A C3FD08            JMP    EXIT



               DISPATCH:
091D DB01              IN     IER
091F E602              ANI    ETBEI   ; ARE TRANSMIT INTERRUPTS ENABLED?
0921 C0                RNZ            ; YES, RETURN
0922 DB05              IN     LSR
0924 E620              ANI    THRE    ; CHECK FOR TX BUFFER EMPTY
0926 C8                RZ             ; WAIT FOR IT TO EMPTY
0927 DB06              IN     MSR
0929 E620              ANI    DSR     ; ONLY SEND NEXT FRAME IF DSR=1
092B C0                RNZ
                       NEXTIN         ; TRY TO GET ANOTHER BUFFER
092C+E7                RST    4
092D C8                RZ             ; RETURN IF NO BUFFERS ARE READY
092E 7E                MOV    A,M     ; GET DATA LENGTH IN BUFFER HEADER
092F 321D10            STA    OUTCOUNT        ; AND SAVE IT FOR INTERRUPT ROUTINE
                       INCRLB 3
0932+3E03              MVI    A,3
0934+DF                RST    3
0935 221A10            SHLD   LBOP    ; LBOP = OLBE + 3
0938 3E03              MVI    A,ETBEI+ERBFI
093A D301              OUT    IER     ; ENABLE TRANSMIT INTERRUPTS
093C C9                RET

               ;************************************************************
               ;       DOWNLOAD PROGRAM
               ;***********************************************************

1800 =         RAM     EQU    1800H   ;START OF PROGRAM RAM


               ;       CHECK FOR RAM VECTORS AT LOC. RAM
               ;       IF VECTORS ARE FOUND GO TO FIRST VECTOR.
               ;       IF NO VECTORS ARE FOUND FLASH LIGHT 6 TIMES AND
               ;       INITIALIZE 8250 FOR DOWNLINE LOADER.
               ;
               ;       WHEN IN THE DOWNLOAD PROGRAM AN '*' WILL
               ;       TERMINATE THE LOADER PROGRAM AND JUMP
               ;       TO FIRST VECTOR AT LOC. RAM


093D 3A0018    OPTION: LDA    RAM     ;CHECK FIRST 4  VECTORS
0940 FEC3              CPI    0C3H
0942 C26009            JNZ    INIT1
```

```
0945 3A0318          LDA     RAM+3
0948 FEC3            CPI     0C3H
094A C26009          JNZ     INIT1
094D 3A0618          LDA     RAM+6
0950 FEC3            CPI     0C3H
0952 C26009          JNZ     INIT1
0955 3A0918          LDA     RAM+9
0958 FEC3            CPI     0C3H
095A C26009          JNZ     INIT1
095D C30018          JMP     RAM     ;GO HERE IF VECTORS PRESENT


                ;    INITIALIZE THE 8250
0960 3E83       INIT1:  MVI     A,83H
0962 D303            OUT     3       ;INIT LINE CNTL REG TO SET DIVISOR
0964 3E10            MVI     A,LOW BAUDRAT
0966 D300            OUT     0
0968 3E00            MVI     A,HIGH BAUDRAT
096A D301            OUT     1
096C 3E03            MVI     A,FORMAT
096E D303            OUT     3       ;INIT LINE CONTROL REG NORMALLY
                ;
                ;  NOW FLAG PROPER RESET CONDITION
0970 CD1D0A          CALL    FLASH   ;FLASH 6 TIMES
                ;
                ;  SET JUMP VECTORS IN RAM
0973 3EC3            MVI     A,0C3H
0975 320018          STA     RAM
0978 320318          STA     RAM+3
097B 320618          STA     RAM+6
097E 320918          STA     RAM+9
0981 211508          LXI     H,TIPINIT
0984 220118          SHLD    RAM+1
0987 213608          LXI     H,RST55
098A 220418          SHLD    RAM+4
098D 210000          LXI     H,0
0990 220718          SHLD    RAM+7
0993 211D09          LXI     H,DISPATCH
0996 220A18          SHLD    RAM+10

0999 CD020A     START:  CALL    CHRIN   ;READ A CHARACTER
099C FE2A            CPI     '*'
099E CA0018          JZ      RAM     ;START OF PROGRAM IN RAM
09A1 FE1A            CPI     1AH     ;CNTR "Z"
09A3 CA0000          JZ      0       ;RESET
09A6 FE3A            CPI     ':'
09A8 C29909          JNZ     START
                ;
09AB CDE509     LOADER: CALL    RBYTE   ;READ BYTE (2 CHARACTERS AND PACK)
09AE FE00            CPI     00      ;EOF?
09B0 CA9909          JZ      START
09B3 5F              MOV     E,A     ;SAVE BYTE COUNT
09B4 57              MOV     D,A     ;SAVE IN CKSUM
09B5 CDE509          CALL    RBYTE   ;GET ADDRESS MSB
09B8 67              MOV     H,A
09B9 82              ADD     D       ;ADD CKSUM
```

```
09BA 57              MOV     D,A
09BB CDE509          CALL    RBYTE   ;READ ADDRESS LSB
09BE 6F              MOV     L,A
09BF 82              ADD     D
09C0 57              MOV     D,A     ;SAVE CKSUM
09C1 CDE509          CALL    RBYTE   ;IGNORE 00 BYTE
09C4 CDE509     LOOP1:  CALL    RBYTE   ;READ DATA BYTE
09C7 77              MOV     M,A     ;LOAD INTO MEMORY
09C8 82              ADD     D
09C9 57              MOV     D,A
09CA 23              INX     H
09CB 1D              DCR     E       ;DECREMENT DATA BYTE COUNTER
09CC C2C409          JNZ     LOOP1
09CF CDE509          CALL    RBYTE   ;READ CKSUM BYTE
09D2 5F              MOV     E,A
09D3 7A              MOV     A,D     ;FETCH COMPUTED CKSUM
09D4 2F              CMA
09D5 3C              INR     A
09D6 BB              CMP     E       ;COMPARE COMPUTED AND READ CKSUM
09D7 C20E0A          JNZ     ERROR
09DA CD020A     LUP2:   CALL    CHRIN   ;LOOK FOR ':' (SKIP CR-LF)
09DD FE3A            CPI     ':'
09DF C2DA09          JNZ     LUP2
09E2 C3AB09          JMP     LOADER
                ;
                ;
09E5 C5         RBYTE:  PUSH    B
09E6 CDF409          CALL    RHEX
09E9 07              RLC
09EA 07              RLC
09EB 07              RLC
09EC 07              RLC
09ED 47              MOV     B,A
09EE CDF409          CALL    RHEX
09F1 B0              ORA     B
09F2 C1              POP     B
09F3 C9              RET
                ;
09F4 CD020A     RHEX:   CALL    CHRIN   ;READ ASCII CHARACTER
09F7 FE3A            CPI     '9'+1
09F9 DAFF09          JC      RDDIG
09FC D637            SUI     'A'-10
09FE C9              RET
09FF D630       RDDIG:  SUI     '0'
0A01 C9              RET
                ;
0A02 DB05       CHRIN:  IN      5       ;READ UART STATUS
0A04 E601            ANI     1
0A06 CA020A          JZ      CHRIN   ;LOOP TILL READY
0A09 DB00            IN      0
0A0B E67F            ANI     7FH     ;ASCII ONLY HERE
0A0D C9              RET
                ;
0A0E 3E40       ERROR:  MVI     A,40H
                     SIM
0A10+30              DB      30H
```

```
0A11 CD300A          CALL    DELAY
0A14 3EC0            MVI     A,0C0H
                     SIM
0A16+30              DB      30H
0A17 CD300A          CALL    DELAY
0A1A C30E0A          JMP     ERROR
                ;
0A1D 0606       FLASH:  MVI     B,6
0A1F 3EC0       LUP1:   MVI     A,0C0H
                     SIM
0A21+30              DB      30H
0A22 CD300A          CALL    DELAY
0A25 3E40            MVI     A,40H
                     SIM
0A27+30              DB      30H
0A28 CD300A          CALL    DELAY
0A2B 05              DCR     B
0A2C C21F0A          JNZ     LUP1
0A2F C9              RET
                ;
0A30 2664       DELAY:  MVI     H,100
0A32 2E00       LUP3:   MVI     L,0
0A34 2D         LUP4:   DCR     L
0A35 C2340A          JNZ     LUP4
0A38 25              DCR     H
0A39 C2320A          JNZ     LUP3
0A3C C9              RET

0A3D                 END
```

I-FRAME de VE3PKT

.---. --.- -

Newsletter of the

HAMILTON & AREA PACKET NETWORK

2391 Arnold Crescent,
Burlington, Ontario, Canada, L7P 4J2

Hank S. Magnuski, KA6M,
311 Stanford Ave.,
Menlo Park, California,
94025.